

1N-35
207546
11P

VLSI-Based Video Event Triggering for Image Data Compression

Glenn L. Williams
Lewis Research Center
Cleveland, Ohio

Prepared for the
40th ISA International Instrumentation Symposium
sponsored by the Instrument Society of America
Baltimore, Maryland

May 15 1994

(NASA-TM-106500) VLSI-BASED VIDEO
EVENT TRIGGERING FOR IMAGE DATA
COMPRESSION (NASA) 11 p

N94-25116

Unclass



G3/35 0207546



VLSI-BASED VIDEO EVENT TRIGGERING FOR IMAGE DATA COMPRESSION

**Glenn L. Williams
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135**

KEYWORDS

Fuzzy logic, Machine vision, Motion analyzers, Personal computer, Display systems, Digital control, Computer control, Video inspection and monitoring; Artificial intelligence

ABSTRACT

Long-duration on-orbit microgravity experiments require a combination of high resolution and high frame rate video data acquisition. The digitized high-rate video stream presents a difficult data storage problem. Data produced at rates of several hundred million bytes per second may require a total mission video data storage requirement exceeding one terabyte. A NASA-designed VLSI-based, highly parallel digital state machine generates a digital trigger signal at the onset of a video event. High capacity random access memory storage coupled with newly available fuzzy logic devices permits monitoring a video image stream for long term (DC-like) or short term (AC-like) changes caused by spatial translation, dilation, appearance, disappearance, or color change in a video object. Pretrigger and post-trigger storage techniques are then adaptable to archiving only the significant video images.

INTRODUCTION

In the late 1990's, NASA will launch several Space Shuttle missions with advanced on-board microgravity experiments. High-speed motion picture film has been used on previous Space Shuttle flights to capture high-rate high-resolution images of the critical portions of transient motions in combustion and fluid experiments in microgravity fields. Motion-picture film must be stored after use, and photographically processed on the ground, which is too late for assisting real-time evaluation and modification of flight experiments. Because of substantial investment in placing an experiment into orbit in the Space Shuttle, it is extremely important to obtain as much scientific data as possible during the

few days of a flight, and real-time modification of a test plan has the potential for yielding valuable data based on iterations in the experiment. High Resolution, High Frame Rate Video Technology is being studied as a possibility for recording and down-linking high-quality images of steady-state and transient motion in microgravity experiments [1]. Video imaging permits ground-based viewing of the experiments in real-time or after short delays for transmission and digital computer processing. Digitized high-rate video immediately presents a difficult data storage problem, because data can be produced at such high rates that total onboard mission video data storage requirements will easily exceed one terabyte. Without careful attention to cost of storage and transmission, such vast volumes of data will become very expensive to support.

The volume and cost of data storage is minimized by hardware which stores only the images of important events, in real-time. These images are acquired only when there is localized motion around some significant physical event in the video scene. In NASA's microgravity experiments, minutes or hours of inactivity may precede significant events. During the waiting period, thousands of redundant video frames can be ignored until something interesting happens.

Using commercially available silicon VLSI circuitry, we have developed and are currently repackaging a system of circuitry which can detect and trigger on motion in a video image stream in less than five milliseconds. The system will support acquisition of many seconds of video frame storage when coupled with high density frame store memory capable of continuously recycling storage used by video frames which have no interesting changes. With pre-trigger and post-trigger capabilities, such memory will store an entire sequence of images including a number of precursor images of any changes visible just before the main event.

With two modes of operation, our Video Event Trigger design can trigger on rapid image changes while ignoring slow changes, or it can trigger on any short or long term total difference from stored static reference images. We will show that the image processing hardware we have developed is an extension of classical FIR filter technology used in one-dimensional waveforms.

BACKGROUND

Research and development in the area of detecting and characterizing motion in the video images has been described in the literature [2],[3],[4],[5].

A human observer, relying on visual observations of the scene or on external devices or externally processed electrical signals, such as from pressure or temperature or acoustic transducers, could create the video event trigger manually. But eye-hand reflex time is far too long to respond to an event in only one or two frame times.

Attempts to generate triggers in software, by analyzing frame-to-frame differences, will not meet the five millisecond response requirements with even the fastest digital processors. A 512-pixel by 480-pixel video image (245,760 pixels), continuously refreshed at 30 frames per second, represents a 7.37 million pixel-per-second data stream of image samples. In order to continue in real-time, two such data streams must be processed in only a few milliseconds. Simple calculations reveal that intelligent frame-to-frame

comparison of 245,760 pixels in only five milliseconds implies a processing capability burst rate of nearly 50 million algorithm loops per second. The detection of "interesting" changes involves noticing changes in color or motion which may often be masked by considerable image clutter and which may require some algorithmic processing of the image to interpret what is happening. Merely subtracting one video frame from another, or looking for motion on the edges of a blob, or calculating the movement of the centroid of the blob are all ways which may fail to generate useful triggers with any one set of software definition of what constitutes interesting changes. With today's technology, no software-based processor capable of the required burst-rate performance could be packaged in a case small enough for inclusion in a microgravity experiment flown on the Space Shuttle.

Digital neural networks initially promise interesting possibilities for image comparison. But neural networks generate consistent decisions only after extensive multiple "training" sessions using "typical" data. Unfortunately, video events are usually characterized as having one-shot, unpredictable changes which are difficult to classify into standard training examples.

For ultimate speed, hardware circuits can be customized to operate faster than software, after comparison algorithms are embedded into custom silicon integrated circuits. For our purposes, we require video event triggering in milliseconds, at relatively low cost, and in a small volume. The system we developed uses a high-degree of parallelism, is semi-autonomous, and relies on hardware-based fuzzy logic comparison techniques with the ability to make incremental or gross corrections to the algorithm on a frame-by-frame basis. We have taken advantage of recent commercial advances in silicon VLSI integrated circuits specially designed to process video data streams using fuzzy logic [9],[10],[11]. The system was breadboarded for testing in a commercial MS-DOS "PC/AT" 286-computer with ISA-bus architecture, without use of custom-designed integrated circuits. We used commonly available TTL integrated circuits, Programmable Array Logic devices (PALs), and one special type of VLSI integrated circuit which was commercially available at the time. The system was packaged on three multilayer printed circuit boards designed on a Mentor Graphics workstation. Most of the small-scale logic spread out on the boards could be repackaged in custom-designed integrated circuits if size becomes a concern. The hardware operates as a large register-based peripheral with a relatively light software involvement for setting registers and processing interrupts. Software execution rate is thus relatively slow even in comparison with the capabilities of the 80286 processor and operating system. All of the frame processing takes place in our circuit boards without image data flowing through the ISA-bus. We have taken advantage of a commercial frame-grabber which has a fast 20 MHz local video bus which is carried from board to board via flat cable technology. The computer software was coded in Microsoft C Version 6.0.

In any system which compares images to detect changes between images, at least two full frames must be available for comparison. The comparison may not begin until after the end of a frame. The net result is that the trigger process lags one or more frame times behind the most recent frame being acquired and occurs concurrently with acquisition of the next frame.

The acquisition process results in video frames being presented sequentially to the Video Event Trigger logic control circuitry where they are captured and stored into temporary memory buffers (1, 2, ..., 5). Buffer 5 holds the oldest frame, buffer 4 the next oldest frame, and so on, with buffer 1 holding the newly acquired frame, the one to be compared to all the others.

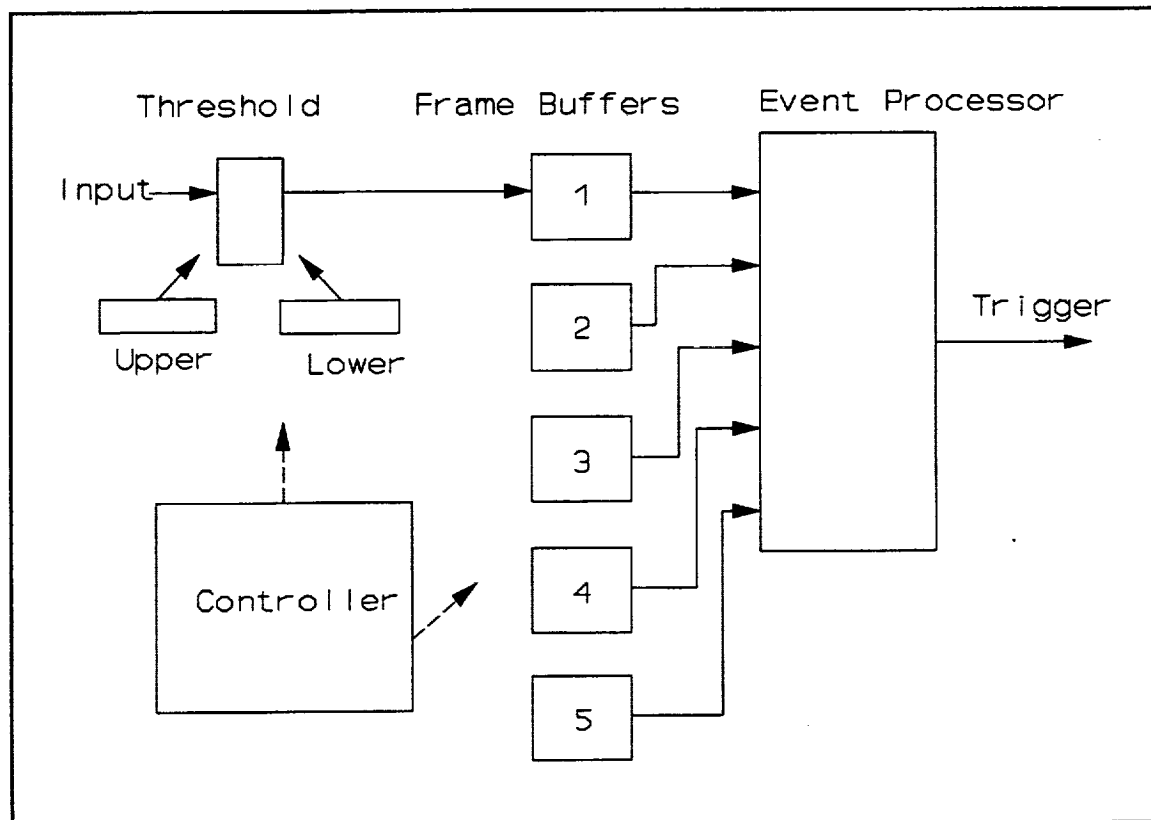


Figure 1 Block diagram of the Video Event Trigger Subsystem.

During system initialization, with the video running continuously, the first few frames are assumed to contain no motion and are stored one-by-one into the k buffers ($k = 5$ in this case), to be used as the "learned" reference frames. Learning in this case merely means loading up the frame buffer memories, one of which can be loaded every frame time.

As previously described, there are two modes for storing and comparing old and new video frames. In the first mode, each new video frame, in buffer 1, is compared against a set of $k-1$ older frames via subtraction and fuzzy logic rules. If no motion is detected, the oldest of the k stored frames is discarded by rearranging pointers to the video frames. Effectively, all the frames are shifted down, and the newest frame is assigned to the first of the reordered $k-1$ frames. Then the next frame is acquired and the cycle is repeated. In this mode very slow changes in the video scene, similar to a slow DC drift in a one-dimensional analog signal, are ignored. Only dramatic changes in the latest video would constitute enough motion to set off a trigger. Operation is thereby very similar in function to "AC Coupling" on an oscilloscope.

In the second mode, new frames are loaded only into buffer 1, and are discarded after use. The remaining $k-1$ stored frames are permanent, non-changing reference frames which were loaded at the

start of the observing run. The method assumes images can contain either slow changes or rapid changes. In this mode, any changes at all are important to the motion detection process, and if they occur, they must be reported when a certain threshold is exceeded. This is similar in function to "DC Coupling" on an oscilloscope.

A MATHEMATICAL BASIS

Our techniques parallel the architectures of one-dimensional FIR and IIR digital waveform filters. However, we avoid the use of recursion, i.e. feeding output images back into the input data stream. We thus avoid problems with instability and limit cycles. But otherwise, our techniques have a close similarity to more commonly digital filter architectures, for which a large pool of documenting literature exists.

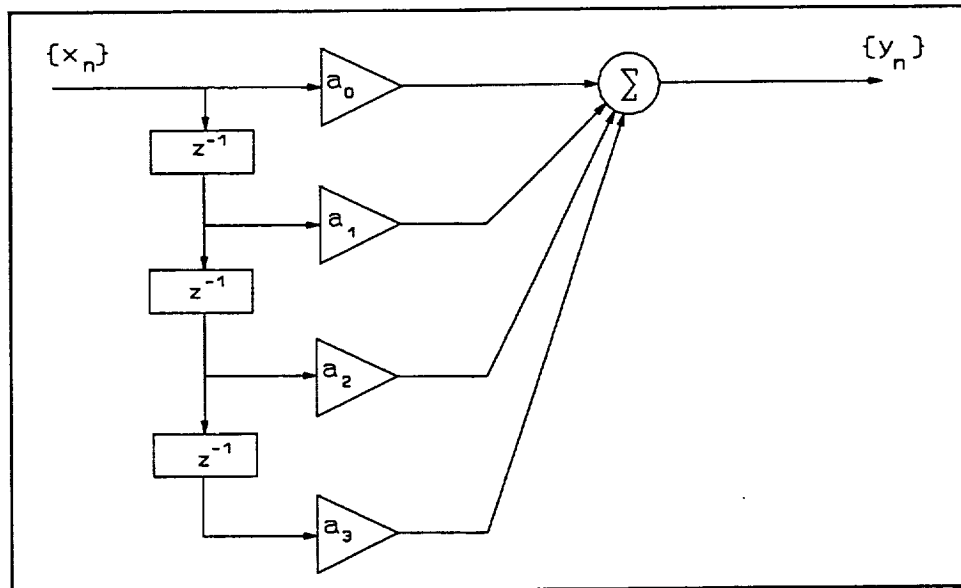


Figure 2 A discrete-time FIR digital filter.

The diagram of Figure 1 is (not altogether accidentally) topologically similar to the diagram, in Figure 2, for the fundamental discrete-time FIR digital filter described in many texts [6][7].

For the FIR filter in Figure 2,

$$y_n = \sum_{k=0}^M a_k x_{n-k} \quad (1)$$

The literature teaches that this filter can be used for processing a previously sampled one-dimensional analog signal in a digital transformation of a particular analog filter. The response of the filter is tuned by the value of the coefficient set $\{a_n\}$. The techniques and theory for calculating the coefficients will not be shown here due to the wide availability of texts on the subject.

We should note here that z^{-1} is the system sample delay, and the summation symbol does not rule out subtraction because the coefficient set $\{a_n\}$ may have one or more negative values.

Over time, each coefficient of the set $\{a_n\}$ becomes a modifier for successively aged copies of the original sample, until after $n+1$ sample times the oldest sample is lost off the end of the chain, for finite n . This filter, then, only processes new samples based on the n older samples, so that a varying ac-like signal $\{x_n\}$ will produce a significant output $\{y_n\}$ if $\{x_n\}$ varies significantly from sample to sample. I.e. an ac-like output $\{y_n\}$ will occur only if $\{x_n\}$ is an ac-like signal.

Assume that at time n , with coefficients $\{a_n\}$, and input $\{x_n\}$, the output $\{y_n\} = \{0\}$.

If the process which updates the chain of successively older samples is halted just after time n , so that no new samples are stored, then at a later time $n+m$, a sample $\{x_{n+m}\}$ will generally be different from $\{x_n\}$, and the output $\{y_{n+m}\}$ will then usually exhibit a constant, or non-zero, output. That is, for any element of the set $\{x_{n+m}\}$, different from the set $\{x_n\}$ which yields $\{y_n\} = \{0\}$, the output set $\{y_{n+m}\}$ will be non-null. This is true even if $\{x_{n+m}\}$ are constant.

Equation 1 was derived for a discrete (i.e. non-fuzzy) process, using one-word sample values from one-dimensional signals, and discrete arithmetic.

With our Video Event Trigger we have extended the theory by empirically showing that:

- (1) The summation can involve fuzzy-logic based arithmetic.
- (2) The sample set $\{x_n\}$ of a one-dimensional signal can be extended to video frames, which are two-dimensional signals.
- (3) The sample recursion interval z^{-1} is the video frame sampling interval of at least 30 frames per second.
- (4) System processing occurs in less than five milliseconds using straight-forward LSI, VLSI, and field-programmable logic packaging.

In our two-dimensional system, using fuzzy logic rules in the event processor, the syntax of the expression for the FIR rules is in need of a little rewriting:

$$y_n = \left(\sum_{k=0}^M \right) a_k(\bullet) x_{n-k} \quad (2)$$

The agreement of theory with practice was predicted before the hardware was built, and works in

practice. But it has not been proven here with mathematical rigor. We propose as a challenge that Equation 2 be proven rigorously and analytically using the rules of fuzzy arithmetic and fuzzy logic. We note that Equation 2 was written here as an extension of Equation 1 using the syntax of fuzzy arithmetic. But we do so without rigorous proof, for our sets $\{x_n\}$ and $\{y_n\}$ represent individual sets of two-dimensional frames operated on by discrete time delays (z-transforms), and (Σ) and (\bullet) represent summation in the rules of fuzzy arithmetic[8] and VLSI-based fuzzy-logic comparators. This results in rather intractable mathematical relationships which we have not attempted to document. Yet we have demonstrated a working system which is intuitively understandable.

FURTHER HARDWARE DETAILS

We enhance the operation and reduce complexity in the event processor by globally thresholding (clipping) the video levels with two digital binary comparators and two "cut" levels. We then have a "window" comparison of the video. "Below Level", "Above Level", "Inside Window", and "Outside Window" are the four choices that result. These levels (and modes) are programmed into local registers on the boards, under computer control. These two levels represent the "alpha-cut" (variable sensitivity) levels that determine which levels of gray (or color attributes) will be reduced to a binary ONE by the comparator. All other levels converted to binary ZERO. Then, after the operator's selective adjustment of the alpha-cut levels, the event processor uses only these clipped images. The processing load is thereby greatly simplified in our design. But doing so is not a requirement in the general case, should a design require full gray-level sensing.

MEMORY REQUIREMENTS

A goal of our design was to minimize the cost of storage of video images. The quantities of data resulting from high frame rate or image resolution conflict with the need for low cost storage.

In a particular example, assume frames of video data can be stored sequentially and cyclically, a frame at a time, in video RAM storage. High speed, large volume RAM memory boards are commercially available and could in theory be modified for this purpose. Assume the existence of a memory controller which makes sure that the storage is cyclic, in such a fashion that the very oldest frame is overwritten (lost) by the newest frame being stored into the memory. We make the "obvious" assumption that the oldest frames carry no data of any value (nothing happened).

Upon an operator "arm" command, the hardware inside the controller starts filling a memory buffer with "pretrigger" data from the digitizer. Once the minimum requirements of the pretrigger buffer have been satisfied, the remaining portion of the buffer is treated as post-trigger data. During the interim, until the trigger signal arrives, the memory is controlled as a wrap-around buffer, in a fashion similar to a continuous-loop magnetic tape recorder. Since the memory has a maximum capacity, the oldest data is continuously replaced with the newest data until the trigger point.

When a video event occurs, the trigger pulse signals the memory controller to begin a new phase of

frame storage algorithms. In this new phase, some of the oldest frames (still redundant) are overwritten by new, interesting frames containing motion. But a selectable number of frames of medium age are retained in memory because they may contain images of precursor activity important to the history leading up to the event. At the trigger time instant, the act of triggering sets a digital logic switch which causes the wrap-around to cease. Thereafter, the post-trigger section of the memory buffer is filled, and then the acquisition of video images into solid-state memory stops. The net effect is that the memory holds the entire useful record of the transient, both before and after the trigger point, depending on the size of the "pretrigger memory" setting. All that is required is a little pointer arithmetic to unwrap the data already in memory (Figure 3.).

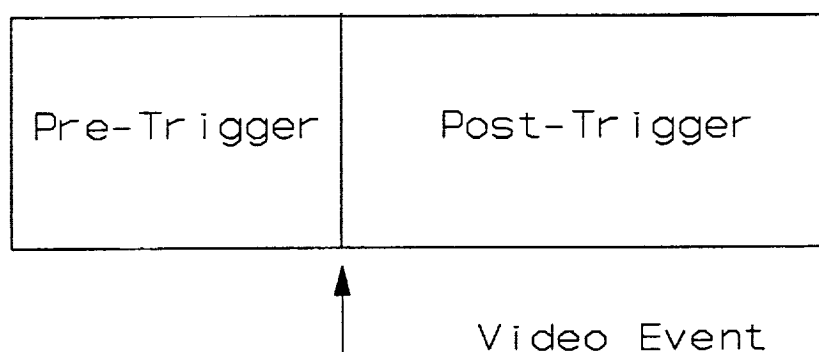


Figure 3 Frame store memory can be divided into pre-trigger and post-trigger portions.

ACKNOWLEDGMENT

This work was supported by NASA Headquarters, Office of Advanced Concepts and Technology.

SUMMARY

We have demonstrated a useful video image acquisition and data-reduction subsystem which we call the Video Event Trigger. This subsystem was implemented with commercially available VLSI integrated circuits which can rapidly process a 20 MHz stream of video data using fuzzy logic rules. The circuit boards were packaged for operation with the industry standard ISA-bus. Dense frame buffer storage memory can be designed to capture a multitude of images before and after the trigger point. The triggering can operate either in the "AC-coupling" or "DC-coupling" mode. We have indicated that one-dimensional FIR digital filter mathematics can be extended to cover both the two-dimensional case and the fuzzy logic case, for interpreting interesting motion in a video image consisting of mostly static information with a localized cluster of moving or changing pixels.

BIBLIOGRAPHY

- [1] High Resolution, High Frame Rate Video Technology, *NASA Conference Publication 3080*, May 11-12, 1988.
- [2] A. K. Jain, "Image Data Compression: A Review," *Proc. IEEE*, Vol. 69, NO. 3, March 1981, pp. 360-362.
- [3] J.R. Jain and A.K. Jain, "Interframe Adaptive Data Compression Techniques For Images," Sig. & Image Proc. Lab., Dep. Elec. and Comput. Eng., Univ. CA, Davis, Aug. 1979.
- [4] A. Rosenfeld and A. Kak, *Digital Picture Processing*, Second Edition, Academic Press, 1982.
- [5] R. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley and Sons, Inc., 1989, pp. 190-192.
- [6] C. Chen, *One-Dimensional Digital Signal Processing*, Marcel Dekker, Inc., 1979, p. 145.
- [7] A. Peled and B. Liu, *Digital Signal Processing*, John Wiley and Sons, Inc., 1976, p. 14.
- [8] A. Kaufman and M. Gupta, *Introduction to Fuzzy Arithmetic, Theory and Applications*, Van Nostrand Reinhold Company Inc., 1985, pp. 222-227.
- [9] A. Kandel, *Fuzzy Techniques in Pattern Recognition*, John Wiley and Sons, Inc., 1978.
- [10] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [11] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE February 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum		
4. TITLE AND SUBTITLE VLSI-Based Video Event Triggering for Image Data Compression		5. FUNDING NUMBERS WU-694-03-OC		
6. AUTHOR(S) Glenn L. Williams				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		8. PERFORMING ORGANIZATION REPORT NUMBER E-8512		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-106500		
11. SUPPLEMENTARY NOTES Prepared for the 40th ISA International Instrumentation Symposium sponsored by the Instrument Society of America, Baltimore, Maryland, May 1-5, 1994. Responsible person, Glenn L. Williams, organization code 4120, (216) 433-2389.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 35			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Long-duration on-orbit microgravity experiments require a combination of high resolution and high frame rate video data acquisition. The digitized high-rate video stream presents a difficult data storage problem. Data produced at rates of several hundred million bytes per second may require a total mission video data storage requirement exceeding one terabyte. A NASA-designed VLSI-based, highly parallel digital state machine generates a digital trigger signal at the onset of a video event. High capacity random access memory storage coupled with newly available fuzzy logic devices permits monitoring a video image stream for long term (DC-like) or short term (AC-like) changes caused by spatial translation, dilation, appearance, disappearance, or color change in a video object. Pretrigger and post-trigger storage techniques are then adaptable to archiving only the significant video images.				
14. SUBJECT TERMS Fuzzy logic; Machine vision; Motion analyzers; Personal computer; Display systems; Digital control; Computer control; Video inspection and monitoring; Artificial intelligence			15. NUMBER OF PAGES 11	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	